IN THE CLAIMS:

Please amend the claims as follows

Claims 1-121 (Cancelled)

122. (New) A design verification apparatus comprising: a verification software and one or more simulators, wherein the said verification software instruments an additional code or a circuit into the design code or into the design net-list in an automatic way, so that the necessary information (with which one or more post-1st simulations that run as the back-stage simulation can be executed against one or more design objects in the said design code or in the design net-list) can be collected during the 1st simulation run, referred to as the front-stage simulation, and the said one or more post-1st simulation runs are executed fast while obtaining the visibility, thereby providing high simulation speed or high visibility for debugging.

123. (New) A design verification method, in which one or more design bugs in the design code or in the design net-list are identified and collected by a number of simulation executions with a number of test benches, comprising the following steps: some of said simulation executions are decomposed into the 1st simulation run as the front-stage simulation and the post-1st simulation runs as the back-stage simulation; verification software instruments an additional code or a circuit into the design code or into the design net-list in an automatic way, so that the minimally necessary information(with which one or more post-1st simulations that run as the back-stage simulation can be executed against one or more design objects in the said design code or in the design net-list) can be collected during the 1st simulation run (the front-stage simulation), and said one or more post-1st simulation runs are executed fast while obtaining the visibility, thereby providing fast simulation speed or high visibility for debugging.

124. (New) A design verification method, in which one or more design bugs in the design code or in the design net-list are identified and corrected by a number of simulation executions with a number of test benches, implemented as follows:

some of the said simulation executions are decomposed into the 1st simulation run as the front-stage simulation and the post-1st simulation runs as the back-stage simulation; verification software instruments an additional code or a circuit into the design code or the design net-list in an automatic way, so that the minimally necessary information(with which one or more post-1st simulations that run as the back-stage simulation can be executed against one or more design objects in the said design code or in the design net-list) can be collected during the 1st simulation run (the front-stage simulation), and said one or more post-1st simulation runs are executed fast while obtaining the visibility, thereby providing both fast simulation speed and high visibility for debugging.

125. (New) A design verification method, in which one or more design bugs in the design code or in the design net-list are identified and corrected by a number of simulation executions with a number of test benches, characterized in that the method is composed of the following steps: the design states of one or more design objects in DUV and, in necessary, the design states of one or more design objects in test bench are saved at one or more simulation times during one or more 1st simulation runs (the front-stage simulation), and the parallel post-1st simulation runs are executed from the saved design states as the back-stage simulation.

126. (New) A design verification method, in which one or more design bugs in the design code or in the design net-list are identified and corrected by a number of simulation executions with a number of test benches, comprising the following steps: some of the said simulation executions are decomposed into the 1st simulation run as the front-stage simulation and the post-1st simulation runs as the back-stage simulation; verification software instruments an additional code or a circuit into the design code or the design net-list in an automatic way so that the necessary information (with which one or more post-1st simulation runs as the back-stage simulation can be executed against one or more design objects in said design code or in the design net-list) can be collected during the 1st simulation run (the front-stage simulation), and the said one or more post-1st simulation runs

are executed fast while obtaining the visibility, thereby providing both fast simulation speed and high visibility for debugging.

127. (New) The design verification method, in which a simulation at the lower level of abstraction is executed rapidly by using the simulation results from one or more simulation runs at the higher level of abstraction.

128. (New) A design verification method according to Claim 127, wherein:

- one or more simulation runs at the higher level of abstraction is a transaction-level simulation and the simulation at the lower level of abstraction is a register transfer-level simulation, a gate-level simulation, a transaction/register transfer mixed-level simulation, a transaction/gate mixed-level simulation, a register transfer/gate mixed-level simulation, or a transaction/register transfer/gate-level mixed-level simulation;

- one or more simulation runs at the higher level of abstraction is a register transfer-level simulation and the simulation at the lower level of abstraction is a gate-level simulation, or a mixed register transfer/gate-level simulation;

- one or more simulation runs at the higher level of abstraction is a simulation based on a simple delay model and the simulation at the lower level of abstraction is a simulation based on a more accurate delay model than the said simple delay model;

- one or more simulation runs at the higher level of abstraction is a cycle-based simulation and the simulation at the lower level of abstraction is an event-driven simulation, or

- the said simulation results from one or more simulation runs at the higher level of abstraction which is used at a simulation at the lower level of abstraction contains either at least design state of one or more design objects so that a simulation at the lower level of abstraction is executed by the temporally parallel execution method or at least one or more input information for replay, or

input/output information for replay of design objects so that a simulation at the lower level of abstraction is executed by the spatially parallel execution method.

129. (New) A design verification apparatus comprising: a verification software and one or more simulators, wherein the said verification software instruments an additional code or a circuit into the design code or into the design net-list in an automatic way so that the necessary information (with which one or more post-1st simulation runs as the back-stage simulation can be executed against one or more design objects in the said design code or in the design net-list) can be collected during the 1st simulation run(the front-stage simulation), and the said one or more post-1st simulation runs are executed fast while obtaining the visibility, thereby providing both fast simulation speed and high visibility for debugging.

130. (New) A design verification method, in which one or more design bugs in the design code or in the design net-list are identified and collected by a number of simulation executions with a number of test benches, comprising the following steps: some of the said simulation executions are decomposed into the 1st simulation run as the front-stage simulation and the post-1st simulation runs as the back-stage simulation; verification software instruments an additional code or a circuit into the design code or into the design net- list in an automatic way so that the necessary information (with which one or more post-1st simulation runs as the back-stage simulation can be executed against one or more design objects in the said design code or in the net-list) can be collected during the 1st simulation run,(the front-stage simulation), and the said one or more post-1st simulation runs are executed fast while obtaining the visibility, thereby providing both fast simulation speed and high visibility for debugging.

131. (New) The design verification method, in which there is no change in the design objects, where the minimal simulation results are saved during the run of the front-stage simulation, the saved simulation results are used for running one or more back-stage simulations either in parallel with two or more simulators or in sequence with a single simulator, and, if necessary, the signal dump for one or

more variables or signals in one or more design objects is carried out during the back-stage simulation, thereby achieving high visibility.

132. (New) The design verification method, in which a simulation at the higher level of abstraction is executed in such a way that its simulation result is entirely or partially corrected by using the simulation result from a simulation at the lower level of abstraction.

133. (New) A design verification apparatus comprising: a verification software and at least two or more different verification platforms, wherein the said verification software instruments the additional code or circuit into the design code or into the net-list in an automatic way so that the necessary information(with which one or more post-1st verification runs as the back-stage verification on the verification platforms among which at least one verification platform is different from the verification platform for the 1st verification run can be executed against one or more design objects in the said design code or net-list) can be collected during the 1st verification run, which is the front-stage verification, and the said one or more post-1st verification runs are executed fast.

134. (New) A design verification method, in which one or more design bugs in the design code or in the design net-list are identified and corrected by running a number of verification executions using at least two or more verification platforms in a hybrid way, comprising the following steps: some of the said verification executions are decomposed into the 1st verification run as the front-stage verification and the post-1st verification runs as the back-stage verification; an additional code or a circuit is instrumented into the design code or the design net-list in an automatic way so that the necessary information (with which one or more post-1st verification runs as the back-stage verification on the verification platforms among which at least one verification platform is different from the verification platform for the 1st verification run can be executed against one or more design objects in the said design code or in the design net-list) can be collected during the 1st simulation run (front-stage simulation), and the said one

or more post-1st simulation runs are executed fast as they are executed against one or more specific blocks only.

135. (New) A design verification method, in which an additional verification uses the results from previous execution of an arbitrary simulation, simulation acceleration, hardware emulation, or prototyping, comprising the following steps" the said additional verification is decomposed into the 1st verification run as the front-stage verification and the post-1st verification runs as the back-stage verification; verification software instruments an additional code or a circuit to inthe design code or into the net-list in an automatic way so that the necessary information (with which one or more post-1st verification runs as the back-stage verification can be executed against one or more design objects in the said design code or in the design net-list) can be collected during the 1st verification run, which is the front-stage verification, and the said one or more post-1st verification runs are executed fast.

136. (New) A design verification apparatus comprising: a verification software and at least one or more verification platforms, wherein the said verification software instruments an additional code or a circuit into the design code or into the net-list in an automatic way so that the dynamic information can be collected during one or more verification runs(simulation runs or simulation acceleration runs), and the said dynamic information collected is re-used at the post-debugging simulation after at least one design object is changed for debugging, thereby entirely or partially reducing total verification time.

137. (New) A design verification method comprising: by using a verification software and at least one or more verification platforms the additional code or circuit is instrumented into the design code or into the design net-list in an automatic way so that the dynamic information can be collected during one or more verification runs(simulation runs or simulation acceleration runs), and the collected dynamic information is re-used at the post-debugging simulation after at least one design object is changed for debugging, thereby entirely or partially reducing total verification time.

138. (New) A design verification method according to Claim 137, wherein the detection method of finding verification time in the case when the verification result of at least one design object changed for debugging differs from the verification result of the said design object before the change, includesto compare either all values of outputs and inouts of the said design object before and after the change, or compare all values of outputs of the said design object before and after the change in an automatic way, or to apply the re-simulation input stimuli(input information for replay), obtained in one or more verification runs(simulation runs or simulation acceleration runs) before modification, to the said changed design object in an automatic way, or to compare either all values of outputs and inouts of the said design object before and after the change or compare all values of outputs of the said design object before and after the change, and to apply the re-simulation input stimuli(input information for replay), obtained in one or more verification runs(simulation runs or simulation acceleration runs) before modification, to the said changed design object in an automatic way.

139. (New) A design verification method according to Claim 137, wherein the verification run(simulation run or simulation acceleration run), after at least one design object has been modified, is executed only either with the said at least one design object changed for debugging and its re-execution input stimuli or with the said at least one design object changed for debugging, its re-execution input stimuli(input information for replay) and only the part of design objects unchanged at least up to the first verification time when the verification result of at least one design object changed for debugging differs from the verification result of the said design object before the change from the verification time 0.

140. (New) A design verification method according to Claim 137, wherein the verification run(simulation run or simulation acceleration run), after at least one design object has been modified, is executed with all the design objects after the first time when the verification result of at least one design object changed for

debugging differs from the verification result of the said design object before the change in an automatic way.

141. (New) A design verification method according to Claim 137, wherein the verification run(simulation run or simulation acceleration run), after at least one design object has been modified, is executed with all the design objects from the time, when the verification result of at least one design object changed for debugging differs from the verification result of the said design object before the change, to the time when the verification result of at least one design object changed for debugging becomes same as the verification result of the said design object before the change in an automatic way.

142. (New) A design verification method according to Claim 137, wherein the verification run(simulation run or simulation acceleration run) after at least one design object has been modified is executed only with the said at least one design object changed for debugging and its re-execution input stimuli(input information for replay) at least to the first verification time when the verification result of at least one design object changed for debugging differs from the verification result of the said design object before the change from the verification time 0 and after the first verification time of different results the verification run(simulation run or simulation acceleration run) is automatically switched to the verification execution with the entire design.

143. (New) A design verification method according to Claim 140, or Claim 142, wherein the verification run(simulation run or simulation acceleration run) with all the design objects at the said first verification time of different results, the restoring design states for the unchanged design objects occurs with the design state information saved during the verification runs(simulation runs or simulation acceleration runs) before the modification for debugging at the particular design checkpoint, which is no later than the said first verification time of different results.

144. (New) A design verification method according to Claim 140, or Claim 142, wherein the verification run(simulation run or simulation acceleration run) with all the design objects at the said first verification time of different results, the restoring design states for the unchanged design objects occurs with the design state information saved during the verification runs(simulation runs or simulation acceleration runs) before the modification for debugging at the particular design checkpoint, which is no later than the said first verification time of different results and be the closest one to the said first verification time of different results.

145. (New) A design verification method according to Claim 142, wherein the verification run(simulation run or simulation acceleration run) after design code modification, the said alignment of the dynamic information of design objects unmodified and modified at the said switching time during the verification run(simulation run or simulation acceleration run) after design code modification, or the said verification run(simulation run or simulation acceleration run) with all design objects after the said switching time after design code modification, is automatically determined by the instrumented code, which is added to the design object, during the verification run(simulation run or simulation acceleration run).